

## CLAIMS

1. A method of dynamically communicating an object message between a client and server of separate object models comprising the steps of:

mapping said client to said server;

5 intercepting a message generated by said client in a first object model;  
examining a second object model for interface information for said server;

generating a translated message for said server; and

forwarding said translated message to said server.

10 2. The method of claim 1 further comprising the step of transmitting a response from said server to said client.

3. The method of claim 1 further comprising the steps of:

said client sending a query to determine if said server is able to respond to said message; and

15 responding affirmatively to said query regardless of whether said server is able to respond to said message.

4. The method of claim 1 wherein said step of mapping further comprises the steps of:

creating a proxy object;

20 creating a stub object; and

creating an association between said proxy object and said stub object.

5. The method of claim 4 further comprising the step of creating an association between said client and said proxy object.

6 The method of claim 4 further comprising the step of creating  
5 an association between said server and said stub object.

7. The method of claim 4 further comprising the step of creating an association between said server and said proxy object.

8 The method of claim 4 further comprising the step of creating an association between said client and said stub object.

10 9. The method of claim 1 wherein said message includes an operation and a plurality of arguments, said step of generating further comprises the steps of:

translating said operation for said server;

translating said plurality of arguments for said server; and

15 generating a translated message using a message protocol of said server.

10. The method of claim 9 wherein said step of translating said arguments further comprises the steps of:

determining the expected number and type of arguments of said

20 server;

determining whether an expected argument type is different than an argument type; and

translating one of said plurality of arguments to an expected argument type when its type is different than said expected argument type.

5           11.    The method of claim 2 wherein said step of transmitting further comprises the steps of:

determining the expected type of said response;

determining whether said type of said response is different than said expected type;

10           → translating said response from said type to said expected type when said type is different than said expected type;

generating a translated response using a message protocol of said client; and

15           transmitting, using said mapping, said translated response to said client.

12.    An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for dynamically handling an object message between a client and server in separate object models, the computer readable program  
20   code in said article of manufacture comprising:

computer readable program code configured to cause a computer to intercept a message generated by said client in a first object model;

computer readable program code configured to cause a computer to examine a second object model for interface information for said server;

5 computer readable program code configured to cause a computer to map said client to said server;

computer readable program code configured to cause a computer to generate a translated message for said server;

10 computer readable program code configured to cause a computer to forward said translated message to said server; and

computer readable program code configured to cause a computer to transmit a response from said server to said object.

13. The article of manufacture of claim 12 further comprising:

15 computer readable program code configured to cause a computer to send a query to determine if said server is able to respond to said message; and

computer readable program code configured to cause a computer to respond affirmatively to said query regardless of whether said server is able to respond to said message.

20 14. The article of manufacture of claim 12 wherein said program code configured to cause a computer to generate a map of said client to said server further comprises:

computer readable program code configured to cause a computer to  
create a proxy object;

computer readable program code configured to cause a computer to  
create a stub object; and

5 computer readable program code configured to cause a computer to  
create an association between said proxy object and said stub object.

15. The article of manufacture of claim 14 further comprising  
computer readable program code configured to cause a computer to create an  
association between said client and said proxy object.

10 16. The article of manufacture of claim 14 further comprising  
computer readable program code configured to cause a computer to create an  
association between said server and said stub object.

15 17. The article of manufacture of claim 14 further comprising  
computer readable program code configured to cause a computer to create an  
association between said server and said proxy object.

18. The article of manufacture of claim 14 further comprising  
computer readable program code configured to cause a computer to create an  
association between said client and said stub object.

20 19. The article of manufacture of claim 12 wherein said message  
includes an operation and a plurality of arguments, said program code

configured to cause a computer to generate a translated message further comprises:

computer readable program code configured to cause a computer to translate said operation for said server;

5 computer readable program code configured to cause a computer to translate said plurality of arguments for said server; and

computer readable program code configured to cause a computer to generate a translated message using a message protocol of said server.

20. The article of manufacture of claim 19 wherein said program  
10 code configured to cause a computer to translate said arguments further comprises:

computer readable program code configured to cause a computer to determine the expected number and type of arguments of said server;

15 computer readable program code configured to cause a computer to determine whether an expected argument type is different than an argument type; and

computer readable program code configured to cause a computer to translate one of said plurality of arguments to an expected argument type when its type is different than said expected argument type.

21. The article of manufacture of claim 12 wherein said program code configured to cause a computer to transmit said response further comprises the steps of:

5 computer readable program code configured to cause a computer to determine the expected type of said response;

computer readable program code configured to cause a computer to determine whether said type of said response is different than said expected type;

10 computer readable program code configured to cause a computer to translate said response from said type to said expected type when said type is different than said expected type;

computer readable program code configured to cause a computer to generate a translated response using a message protocol of said client; and

15 computer readable program code configured to cause a computer to transmit, using said mapping, said translated response to said client.

22. A dynamic object message broker comprising:

a first computer system having a first object model and a first object running in said first object model;

20 an mediating component coupled to said first computer system, said mediating component capable of creating a dynamic messaging interface; and

a second computer system coupled to said mediating component, said second computer system having a second object model and a second object running in said second object model.

23. The message broker of claim 22 wherein said mediating  
5 component comprises:

a control module, said control module capable of creating a mapping between said client object and said server object;

a proxy object coupled to said a control module; and  
a stub object coupled to said proxy object.

10 24. The message broker of claim 23 wherein said first object is a client object, and said proxy object is coupled to said client object.

25. The message broker of claim 24 wherein said second object is a server object, and said stub object is coupled to said server object.

15 26. The message broker of claim 22 further comprising a second mediating component coupled to said mediating component.

27. A method of dynamically communicating an object message between a client and server in separate object models comprising the steps of:

intercepting a message generated by said client in a first object model;



examining a second object model for interface information for said server;

creating a proxy object associated with said client;

creating a stub object associated with said server;

5 determining a message protocol for said server;

generating a translated message from said message using said message protocol; and

forwarding said translated message to said server.

28. The method of claim 27 further comprising the step of creating  
10 an association between said proxy object and said stub object.

29. The method of claim 27 wherein said message includes an operation and a plurality of arguments, said method further comprising the steps of:

translating said operation for said server; and

15 translating said plurality of arguments for said server.

30. The method of claim 27 wherein said step of translating said arguments further comprises the steps of:

determining the expected number and type of arguments of said server;

20 determining whether an expected argument type is different than an argument type; and

translating one of said plurality of arguments to an expected argument type when its type is different than said expected argument type.

31. A method of dynamically handling a message between separate object models comprising the steps of:

- 5        intercepting a message generated by a client in a first object model;
- examining a plurality of second object models to locate a server to process said message;
- obtaining interface information for said server running in one of said plurality of second object models;
- 10        generating a translated message for said server; and
- forwarding said translated message to said server.